

Efficient MCMC for parameter inference for Markov jump processes

Boqian Zhang

Department of Statistics, Purdue University

Vinayak Rao

Department of Statistics, Purdue University

April 11, 2017

Abstract

Markov jump processes (MJPs) are continuous-time stochastic processes that find wide application in a variety of disciplines. Inference for MJPs typically proceeds via Markov chain Monte Carlo, the state-of-the-art being an auxiliary variable Gibbs sampler proposed recently in [14]. This algorithm was designed for the situation where the MJP parameters are known, and Bayesian inference over unknown parameters is typically carried out by incorporating this into a larger Gibbs sampler. This strategy of alternately sampling parameters given path, and then path given parameters can result in poor Markov chain mixing. In this work, we propose a simple and elegant algorithm to address this problem. Our scheme brings Metropolis-Hastings (MH) approaches for discrete-time hidden Markov models (HMMs) to the continuous-time setting, and also also ties up some of the loose ends in [14]. The result is a complete and clean recipe for parameter and path inference in MJPs. In our experiments, we demonstrate superior performance over the Gibbs sampling approach, as well as other approaches like particle Markov chain Monte Carlo [1].

Keywords: Markov jump process, Markov chain Monte Carlo, Metropolis-Hasting, Bayesian inference, Uniformization

1 Introduction

Markov jump processes (MJPs) are continuous-time stochastic processes that have found wide application in fields like computational chemistry [6], population genetics [5], mathematical finance [4], queuing theory [2], artificial intelligence [15] and social-network analysis [11]. MJPs

have been used to model the state of a chemical reaction, the state of a queuing network, segmentation of a strand of DNA, user activity on social media, among many others.

MJPs model temporal evolution in continuous-time, resulting in realistic, mechanistic, and interpretable models, often amenable to mathematical analysis. These same dynamics however raise computational challenges in statistical applications, where given partial and noisy measurements, one has to make inferences over the latent MJP trajectory as well as any system parameters. Such inference is complicated by two facts: one cannot *a priori* bound the number of state transitions, and the state-transition times themselves are continuous-valued. This is in contrast to the situation with *discrete-time* hidden Markov models, and trajectory inference for MJPs typically proceeds via Markov chain Monte Carlo. The state-of-the-art is a recent auxiliary variable Gibbs sampler proposed in [14], we will henceforth refer to this as the Rao-Teh algorithm. The Rao-Teh algorithm was designed to sample paths when the MJP parameters are known. Parameter inference is typically carried out by incorporating this into an outer Gibbs sampler that also simulates parameters given the currently sampled trajectory.

In many situations, the MJP trajectory and parameters can exhibit strong coupling, so that a Gibbs sampler that alternately samples path given parameters, and then parameters given path can mix poorly. In this work, we propose a Metropolis-Hastings framework to address this issue. Our proposed solution is simple and elegant, additionally, it ties up some of the loose ends in the Rao-Teh algorithm. In our experiments, we demonstrate superior performance over Gibbs sampling, as well as other approaches like particle Markov chain Monte Carlo [1].

2 Markov jump processes (MJPs)

Formally, a Markov jump process [3] is a right-continuous piecewise-constant stochastic process $S(t)$ taking values in a countable, and usually finite state space \mathcal{S} (see Figure 2, top-left). For simplicity, we will assume N -states, with $\mathcal{S} = \{1, \dots, N\}$. Then, an MJP is parameterized by two quantities, an N -component probability vector π and a rate-matrix A . The former gives the distribution over states at the initial time (which without loss of generality we assume is 0), while the latter is an $N \times N$ -matrix governing the dynamics of the system. An off-diagonal element A_{ij} , for some $i \neq j$ gives the rate at which the system transitions from state i to j . We write A_i for the negative of the i th diagonal element A_{ii} . For an MJP, $A_i = -A_{ii} = \sum_{j \neq i} A_{ij}$, so that the

rows of A sum to 0. A_i is the absolute value of the diagonal, and gives the total rate at which the system leaves state i for any other state. To simulate an MJP trajectory over an interval $[0, \mathcal{T}]$, one follows Gillespie's algorithm [6]: first sample an initial state s_0 from the distribution π , and then defining $t_0 = t_{curr} = 0$ and $k = 0$, repeat the following two steps while $t_{curr} < \mathcal{T}$:

- Sample a wait-time Δt_k from an exponential distribution with rate A_{s_k} . Set $t_{k+1} = t_{curr} = t_k + \Delta t_k$. The MJP remains in state s_k until time t_{k+1} .
- At the end of this time, jump to a new state $s_{k+1} \neq s_k$ with probability equal to $A_{s_k s_{k+1}} / A_{s_k}$. Set $k = k + 1$.

The set of times of $T = (t_0, \dots, t_{|T|})$ and states $S = (s_0, \dots, s_{|T|})$ together define the MJP trajectory $S(t)$.

2.1 Structured rate matrices

In general, the rate matrix A has $N(N - 1)$ free parameters, giving transition rates between every distinct pair of states. In typical applications, especially when large state-spaces are involved, this $N \times N$ matrix is determined by a much smaller set of parameters. We will write these as θ , with A a deterministic function of these parameters: $A \equiv A(\theta)$. The parameters θ are often more interpretable than the elements of A and correspond directly to physical, biological or environmental parameters of interest. We give three examples below:

The immigration-death process This is a simple MJP governed by two parameters: an arrival rate α and a death-rate β . The state space \mathcal{S} represents the size of a population or the capacity of a queue. New individuals enter according to a rate- α Poisson process, so that the off-diagonal elements $A_{i,i+1}$ all equal α . On the other hand, each individual dies (or each job completes) at a rate β , so the system moves from state i to $i - 1$ with rate $A_{i,i-1} = i\beta$. All other transitions have rate 0, so that $\theta = (\alpha, \beta)$, and $A(\theta)$ is a tri-diagonal matrix.

Birth-death processes This is a simple variant of the immigration-death process where the system moves from state i to $i + 1$ with rate $i\alpha$, so that the population grows at a rate proportional to the population size. Once again, $\theta = (\alpha, \beta)$.

Codon substitution models Such models are used in genetics to characterize transition-rates between nucleotides or codons at a locus on a DNA or RNA molecule over evolutionary time. In the simplest case, all transitions have the same rate [9], and the model is characterized by a single parameter. Other models categorize transitions into groups, for instance synonymous transitions encoding the same amino acid, and nonsynonymous transitions encoding different amino acids. These transitions types have their own rates, and noting that there are 61 amino acids, synonymous/nonsynonymous model results in a 61×61 transition matrix determined by 2 parameters. More refined models [7] introduce additional parameters, however the number of parameters is still significantly smaller than the general case.

3 Bayesian inference for MJPs

In practical situations, an MJP trajectory is only observed a finite set of times, and typically, these observations themselves are noisy. There are then two questions than the practitioner faces:

- What is the MJP trajectory underlying the observations?
- What are the unknown parameters governing the dynamics of the latent MJP?

3.1 Trajectory inference for MJPs

This problem was addressed in [14], and was extended to a broader class of MJPs (as well as other jump processes like semi-Markov processes) in [13]. Both schemes center on alternate approaches to Gillespie’s algorithm, which introduce auxiliary *candidate* jump times that are *thinned* while simulating an MJP trajectory. We focus on the simpler, more widely used algorithm from [14], which is based on an idea called *uniformization* [8]. We refer to this as the Rao-Teh algorithm.

Recall that the diagonal element A_i of the rate matrix give the rate at which the MJP leaves state i for any other state. Importantly, parameters are set up so that self-transitions cannot occur. Now introduce an additional parameter $\Omega \geq \max_i A_i$; [14] suggest setting $\Omega = 2 \max_i A_i$. Instead of sequentially sampling a wait-time and then a new state as in Gillespie’s algorithm, we first simulate a set of candidate transition-times over the interval $[0, \mathcal{T}]$. We draw these from a homogeneous Poisson process with rate Ω . Call these times W ; these along with 0 define a

random grid on $[0, \mathcal{T}]$. Define $B = (I + \frac{1}{\Omega}A)$; observe that this is a stochastic matrix with positive elements, and rows adding up to 1. Assign state-values to the elements in $0 \cup W$ according to a discrete-time Markov chain with initial distribution π , and transition matrix B . Call these states V . Thus $v_0 \sim \pi$, while $p(v_{k+1} = j | v_k = i) = B_{ij}$ for $k \in \{0, \dots, |W| - 1\}$. Note that $\Omega > \max_i A_i$ results in more candidate-times than actual MJP transitions; at the same time the transition matrix B allows self-transitions (unlike A). These two effects cancel each other out, and trajectories sampled this way for any $\Omega \geq \max_i A_i$ have the same distribution as trajectories sampled by Gillespie's algorithm [8, 14].

Introducing the thinned variables allowed [14] to develop a novel and efficient MCMC sampler. We outline this in algorithm 1. At a high-level, the algorithm proceeds by alternately sampling a new grid W conditioned on the MJP trajectory $S(t)$, and then a new trajectory $S(t)$ conditioned on the grid W . The latter step can be carried out using standard techniques from the discrete-time HMM literature. [14] show that the resulting Markov chain targets the desired posterior distribution over trajectories, and is ergodic for any choice of Ω strictly greater than all the A_i 's. As mentioned earlier, they suggest setting $\Omega = 2 \max_i A_i$.

Algorithm 1 The Rao-Teh algorithm [14]: an auxiliary variable sampler for MJP trajectories

Input: MJP parameters θ ; a set of partial and noisy observations X .
A parameter $\Omega > \max_i A_i$, where $A = A(\theta)$ is the MJP rate-matrix.
The previous MJP path $S(t) = (S, T)$.

Output: A new MJP trajectory $\tilde{S}(t) = (\tilde{S}, \tilde{T})$.

- 1: **Given the MJP trajectory (S, T) , sample a new set of thinned candidate times U :**
These are distributed as an inhomogeneous Poisson process with intensity $\Omega - A_{S(t)}$. Since the intensity is piecewise-constant, simulating it is straightforward.
 - 2: **Given the thinned and actual transition times $W = (T \cup U)$ from the previous iteration (after discarding state information S), sample a new trajectory:** Conditioned on the skeleton W , the set of candidate jump times is fixed, and trajectory inference reduces to inference for the familiar discrete-time hidden Markov model (HMM) with initial distribution π , and transition matrix B . [14] use the forward-filtering backward-sampling (FFBS) algorithm: this is an efficient dynamic programming algorithm that makes a forward pass through the finite set of candidate times, sequentially updating the distribution over states at each time $w \in W$. Between any two consecutive elements of W , the system remain in a fixed state, with the likelihood for a state i equal to the likelihood under state i of all observations falling in that interval. At the end of the forward pass, we have a distribution over states at the end time that accounts for all observation. The algorithm then makes a backward pass through the times in W , sequentially sampling the state at any time given the state at the following time, and a distribution over states calculated during the forward pass.
-

3.2 Parameter inference for MJPs

In practice, the MJP parameters themselves are unknown: often, these are the quantities of primary interest when studying a dynamical system. A Bayesian approach places a prior $p(\theta)$ over these unknown variables, and the resulting posterior distribution $p(\theta|X)$ is approximated with samples drawn by Gibbs sampling. In particular, for an arbitrary initialization of the parameters and the trajectory, one repeats the following two steps:

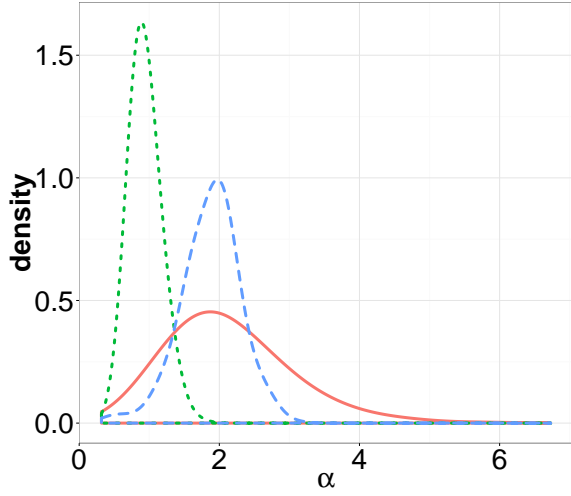


Figure 1: Prior distribution of an MJP parameter (the wide red histogram), as well as two conditional distributions. The narrow green histogram is the distribution conditioned on both the observations as well as a simulated MJP posterior. The wider blue histogram is the distribution of interest: the marginal distribution of the parameters conditioned on observations. These plots were produced from the experiment in section 6.3.

Algorithm 2 Gibbs sampling for parameter inference for MJPs

Input: A set of partial and noisy observations X ,

The previous MJP path $S(t) = (S, T)$, the previous MJP parameters θ .

Output: A new MJP trajectory $\tilde{S}(t) = (\tilde{S}, \tilde{T})$, new MJP parameters $\tilde{\theta}$.

- 1: Sample a trajectory from the conditional $p(S_{new}(t)|X, S_{curr}(t), \theta_{curr})$ following algorithm 1.
 - 2: Sample a new θ from the conditional $p(\theta_{new}|X, S_{new}(t))$.
-

The last distribution depends on a set of sufficient statistics of the MJP trajectory: how much time is spent in each state, and the number of transitions between each pair of states. In special circumstances, θ can be directly sampled from its conditional distribution, otherwise, one has to use a Markov kernel like Metropolis-Hastings or Hamiltonian Monte Carlo to update θ from the conditional $p(\theta_{new}|X, S(t), \theta_{curr})$. In any case, the Gibbs sampling approach of algorithm 2 comes with a well-known limitation: coupling between path and parameters can result in a very sluggish exploration of parameter and path space. We illustrate this in figure 1, which shows the posterior distribution of an MJP parameter in blue (this is the distribution of interest, conditioned only

on the observations) along with the distribution conditioned on both the observations as well as a realization of the MJP trajectory (in green). We have also included the prior distribution in red. Observe how much more concentrated the latter is compared to the former. The coupling is strengthened as the trajectory grows longer and longer, and the Gibbs sampler can mix very poorly for situations with long observation periods, even if the observations themselves are sparse and only mildly informative about the parameters.

For the discrete-time case, this problem of parameter-trajectory coupling can be circumvented by marginalizing out the MJP trajectory and directly sampling from the posterior over parameters $p(\theta|X)$. In its simplest form, this approach involves a Metropolis-Hastings scheme that proposes a new parameter θ from some proposal distribution $q(\theta_{new}|\theta_{old})$, accepting or rejecting according to the usual Metropolis-Hastings probability. The latter step requires calculating the marginal probability of the observations $p(X|\theta)$, integrating out the exponential number of possible latent trajectories. Fortunately this marginal probability is a by-product of the forward-backward algorithm used to sample a new trajectory, so that no additional computational burden is involved. The overall algorithm then is:

Algorithm 3 Metropolis-Hastings parameter inference for a discrete-time Markov chain

Input: A set of partial and noisy observations X ; a proposal distribution $q(\theta^*|\theta)$.
The previous Markov chain parameters θ .

Output: A new Markov chain parameter θ^* .

- 1: Propose a new parameter θ^* from the proposal distribution $q(\theta^*|\theta)$.
 - 2: Run the forward pass of the forward-backward algorithm to obtain the marginal likelihood of the observations, $p(X|\theta^*)$.
 - 3: Accept the proposed θ^* according to the MH probability, $\min(1, \frac{p(X, \theta^*)q(\theta|\theta^*)}{p(X, \theta)q(\theta^*|\theta)})$.
 - 4: If desired, a new trajectory sample can be obtained by completing the backward pass of the forward-backward algorithm for the chosen parameter.
-

3.3 A marginal sampler for MJP parameters

Constructing such a marginal sampler over the MJP parameters by integrating out the continuous-time hidden trajectory is less straightforward: the set of transition times is unbounded, with individual elements unconstrained over the observation interval $[0, \mathcal{T}]$. One approach [5] is to

instead make a sequential forward pass through all *observations* X , using the matrix-exponential operator to marginalize out the infinite number of possible continuous-time trajectories linking two successive times. As demonstrated in [14] however, this approach has a number of drawbacks: it scales cubically rather than quadratically with the number of states, it cannot exploit structure like sparsity in the transition matrix, and can depend in not trivial ways on the exact nature of the observation process. Additionally, the number of expensive matrix exponential calculations scales with the number of observations rather than the number of transitions the system makes.

A second approach is particle MCMC [1]. Here, one uses particle filtering to obtain an unbiased estimate of the marginal probability $p(X|\theta)$; this is then plugged into the Metropolis-Hastings acceptance probability. While the resulting MCMC sampler targets the correct posterior distribution, the resulting scheme does not exploit the structure of the MJP, and we show that it can be quite inefficient.

Work in [14, 13] demonstrated the advantage of introduced the thinned events U : this allows exploiting discrete-time algorithms like FFBS for efficient parameter inference. In the next section, we outline a naïve first attempt at extending this approach to parameter inference. We describe why this approach is not adequate, and describe our final algorithm in the following section.

4 Naïve parameter inference via Metropolis-Hastings

The key idea of the Rao-Teh algorithm [14] is to introduce a set of thinned candidate transition times U from a rate- $(\Omega - A_{S(t)})$ Poisson process. These, along with the extant transition times T , form a random grid W , conditioning on which sampling a new trajectory reduces to a standard discrete-time HMM sampling step. This suggests conditioning on the random grid to update the MJP parameters as well, following the discrete-time MH-scheme from algorithm 3. In particular, we propose a new parameter θ^* from $q(\theta^*|\theta)$, now conditioning on the set of times W . We then make a forward pass over W , and calculate the marginal probabilities $p(X|W, \theta)$ and $p(X|W, \theta^*)$. These can be used to calculate the MH-acceptance probability $\min\left(1, \frac{p(X|W, \theta^*)p(W|\theta^*)p(\theta^*)q(\theta|\theta^*)}{p(X|W, \theta)p(W|\theta)p(\theta)q(\theta^*|\theta)}\right)$. After accepting or rejecting θ^* , the new θ can be used in a backward pass that samples a new trajectory. We then discard all self-transitions and repeat; figure 2 sketches out this scheme.

The resulting algorithm updates θ with the MJP trajectory integrated out, and one would expect it to mix more rapidly than simple Gibbs sampling. It is important to note however that θ

is updated conditioned on W , and that the distribution of W depends on θ . These are the $p(W|\theta)$ terms in the acceptance probability; under uniformization, W follows a homogeneous Poisson process with rate $\Omega(\theta) = 2 \max A(\theta)$. The fact that the MH-acceptance probability involves a $p(X|\theta)$ term is inevitable, however in our experiments, we found that the $p(W|\theta)$ terms have a significant effect acceptance probability. Any proposal that halves $\max A_s$ (and thus Ω) will halve the mean and variance of the distribution of the number of events in W , resulting in an extremely low acceptance probability. In the next section, we describe a way around this issue.

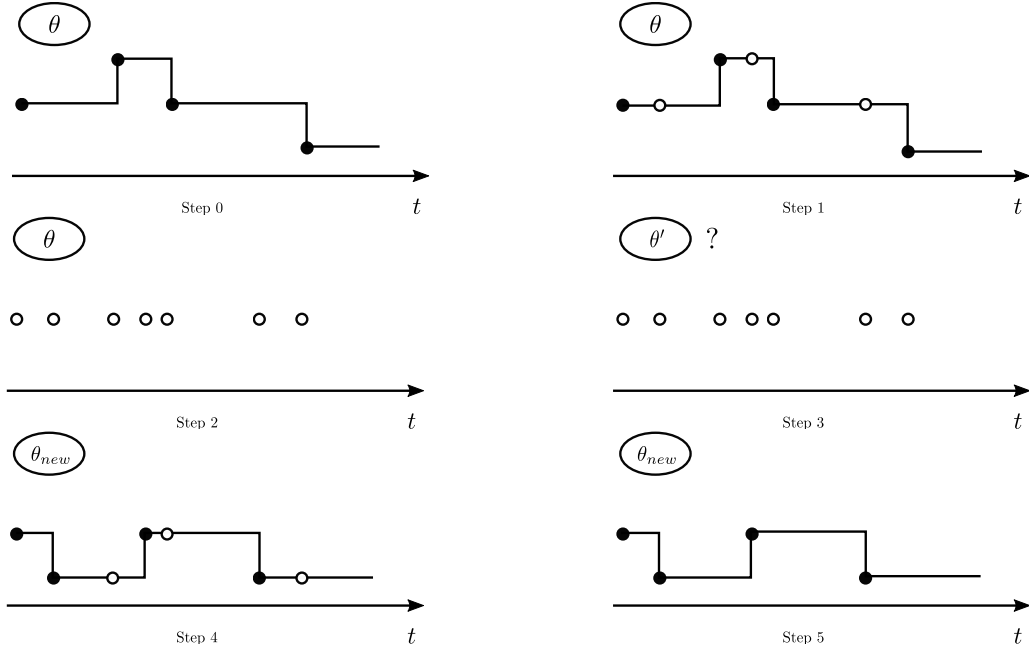


Figure 2: Naïve MH-algorithm. Step 0 to 2: sample thinned events and discard state information to get a random grid. Step 3: propose a new parameter θ' , and accept or reject by making a forward pass on the grid. Steps 4 to 5: make a backward pass using the accepted parameter and discard self-transitions to produce a new trajectory.

Algorithm 4 Naïve MH for parameter inference for MJPs

Input: A set of partial and noisy observations X .

The previous MJP path $S(t) = (S, T)$, the previous MJP parameters θ .

A Metropolis-Hasting proposal $q(\cdot|\theta)$.

Output: A new MJP trajectory $\tilde{S}(t) = (\tilde{S}, \tilde{T})$, new MJP parameters $\tilde{\theta}$.

- 1: Set $\Omega \equiv \Omega(\theta) > \max_s A_s(\theta)$ for some deterministic function $\Omega(\cdot)$ (e.g. $\Omega(\theta) = 2 \max_s A_s(\theta)$).
- 2: Sample virtual jumps $U \subset [t_{start}, t_{end}]$ from a nonhomogeneous Poisson process with piecewise-constant rate $R(t) = (\Omega - A_{S(t)})$. Define $W = T \cup U$ and discard all MJP state information.
- 3: Propose $\theta^* \sim q(\cdot|\theta)$. The acceptance probability is given by

$$\alpha = 1 \wedge \frac{p(W, \theta^*|y) q(\theta|\theta^*)}{p(W, \theta|y) q(\theta^*|\theta)} = 1 \wedge \frac{p(y|W, \theta^*)p(W|\theta^*)p(\theta^*)}{p(y|W, \theta)p(W|\theta)p(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}.$$

- 4: For both θ and θ^* , make a forward pass through the elements of W , sequentially updating the distribution over states at $w \in W$ given observations upto w . For any θ , the Markov transition matrix $B(\theta)$ equals $I + \frac{A(\theta)}{\Omega(\theta)}$ while the initial distribution over states is π_0 . The likelihood of state s at step i is $L_i(s) = p(Y_{[w_i, w_{i+1})}|S(t) = s, t \in [w_i, w_{i+1})) = \prod_{o: t_o \in [w_i, w_{i+1})} p(y_{t_o}|s)$. At the end, we have $p(X|W, \theta)$ and $p(X|W, \theta^*)$. Use these, and the fact that $p(W|\theta)$ is Poisson-distributed to accept or reject the proposed θ and θ^* . Write the new state space as $(W, \tilde{\theta}, \tilde{\theta}^*)$.
 - 5: For the new parameter $\tilde{\theta}$, make a backward pass through the elements of W , sequentially assigning a state to each element of W . This completes the FFBS algorithm.
 - 6: Let \tilde{T} be the set of times in W when the Markov chain changes state. Define \tilde{S} as the corresponding set of state values. Return $(\tilde{S}, \tilde{T}, \tilde{\theta})$.
-

5 An improved Metropolis-Hasting algorithm

Our main idea is to symmetrize the probability of W under the old and new θ , so that the term $p(W|\theta)$ disappears from the acceptance probability. This will result in a simpler, and significantly more efficient MCMC scheme.

As before, the MCMC iteration begins with the pair $(S(t), \theta)$. Instead of simulating the thinned events U , we first generate a new parameter θ^* from some distribution $q(\theta^*|\theta)$. Treat this as an auxiliary variable, so that the augmented space now is the triple $(S(t), \theta, \theta^*)$. We

now pretend $S(t)$ was sampled by a uniformization scheme where the dominating Poisson rate is given by $(\Omega(\theta) + \Omega(\theta^*))$ instead of just $\Omega(\theta)$ (any choice greater than $\max_i A_i$ is valid). It now follows that the set of thinned events U is a piecewise-constant Poisson process with intensity $\Omega(\theta) + \Omega(\theta^*) - A_{S(t)}$. Following [14], the set W , the union of these events with the actual trajectory transition times T , is a realization of homogeneous Poisson process with rate $\Omega(\theta) + \Omega(\theta^*)$. Now we discard all MJP state information, so that the MCMC state space consists of W , the current MJP parameter θ , and the auxiliary parameter θ^* . Finally, we make an MH proposal that swaps θ with θ^* . Observe from symmetry that the Poisson skeleton W has the same probability both before and after this proposal, so that unlike the previous scheme, the ratio $p(W|\theta^*)/p(W|\theta)$ does not appear in the acceptance ratio. This simplifies computation, and significantly improves mixing. The acceptance probability is given by

$$\text{acc} = \min \left(1, \frac{p(X, \theta^*)q(\theta|\theta^*)}{p(X, \theta)q(\theta^*|\theta)} \right) = \min \left(1, \frac{p(X|\theta^*)p(\theta^*)q(\theta|\theta^*)}{p(X|\theta)p(\theta)q(\theta^*|\theta)} \right).$$

The terms $p(X|\theta^*)$ and $p(X|\theta)$ can be calculated by running a forward pass of the forward-backward algorithm. Having accepted or rejected the proposal, a new trajectory is sampled by completing the backward pass, after which the thinned events are discarded. We sketch out our algorithm in figure 3 and algorithm 5.

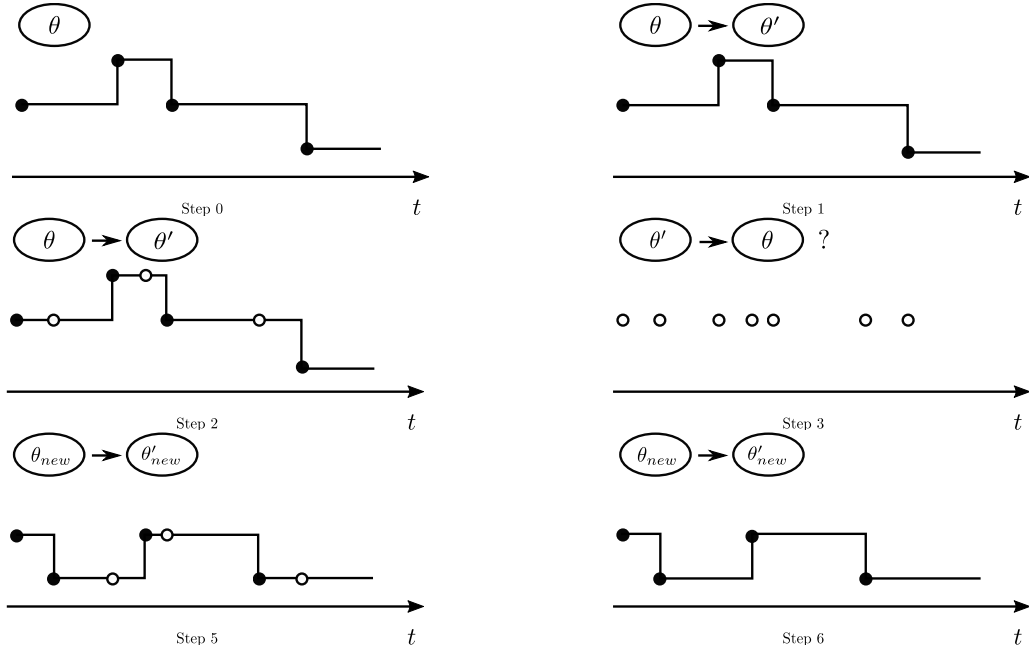


Figure 3: Steps 0-3: Starting with a trajectory and parameter θ , simulate an auxiliary parameter θ' , and then the thinned events U from a rate $\Omega(\theta) + \Omega(\theta') - A_{S(t)}$ Poisson process. Step 4: Propose swapping θ and θ' . Step 5: Run a forward pass to accept or reject this proposal, and use the accepted parameter to simulate a new trajectory. Step 6: Discard the thinned events.

Algorithm 5 Improved MH for parameter inference for MJPs

Input: A set of partial and noisy observations X .

The previous MJP path $S(t) = (S, T)$, the previous MJP parameters θ .

A Metropolis-Hasting proposal $q(\cdot|\theta)$.

Output: A new MJP trajectory $\tilde{S}(t) = (\tilde{S}, \tilde{T})$, new MJP parameters $\tilde{\theta}$.

- 1: Sample $\theta^* \sim q(\cdot|\theta)$, and set $\Omega = \max_i A_i(\theta) + \max_i A_i(\theta^*)$.
- 2: Sample virtual jumps $U \subset [t_{start}, t_{end}]$ from a nonhomogeneous Poisson process with piecewise-constant rate $R(t) = (\Omega - A_{S(t)}(\theta))$. Define $W = T \cup U$ and discard all MJP state information.
- 3: The current MCMC state-space is (W, θ, θ^*) . Propose swapping θ and θ^* , so that the new state-space is (W, θ^*, θ) . The acceptance probability is given by

$$\alpha = 1 \wedge \frac{p(X|W, \theta^*)p(\theta^*)q(\theta|\theta^*)}{p(X|W, \theta)p(\theta)q(\theta^*|\theta)}.$$

- 4: For both θ and θ^* , make a forward pass through the elements of W , sequentially updating the distribution over states at $w \in W$ given observations upto w . At the end, we have calculated $p(X|W, \theta)$ and $p(X|W, \theta^*)$. Use these to accept or reject the proposed swapping of θ and θ^* . Write the new state space as $(W, \tilde{\theta}, \tilde{\theta}^*)$.
 - 5: For the new parameter $\tilde{\theta}$, make a backward pass through the elements of W , sequentially assigning a state to each element of W .
 - 6: Let \tilde{T} be the set of times in W when the Markov chain changes state. Define \tilde{S} as the corresponding set of state values. Return $(\tilde{S}, \tilde{T}, \tilde{\theta})$.
-

Proposition 1. *The sampler described in Algorithm 5 has the posterior distribution $p(\theta, S(t)|X)$ as its stationary distribution.*

Proof. Suppose that at the start of the algorithm, we have a pair $(\theta, S(t))$ from the posterior distribution $p(\theta, S(t)|X)$. Introducing θ^* from $q(\theta^*|\theta)$ results in a triplet whose marginal over the first two variables is still $p(\theta, S(t)|X)$.

Sampling U from a Poisson process with rate $\Omega(\theta) + \Omega(\theta^*) - A_{S(t)}(\theta)$, results in a random grid $W = T \cup U$ that is distributed according to a rate $\Omega(\theta) + \Omega(\theta^*)$ Poisson process (Proposition 2 in [14]). Discarding all state information results in a triplet (W, θ, θ^*) with probability proportional to $p(\theta)q(\theta^*|\theta)p(W|\theta, \theta^*)p(X|W, \theta, \theta^*)$.

Next we propose swapping θ and θ^* , since this is a deterministic proposal, the MH-acceptance probability is given by

$$\alpha = 1 \wedge \frac{p(\theta^*)q(\theta|\theta^*)p(W|\theta^*, \theta)p(X|W, \theta^*, \theta)}{p(\theta)q(\theta^*|\theta)p(W|\theta, \theta^*)p(X|W, \theta, \theta^*)}$$

The term $p(W|\theta, \theta^*)$ is just a Poisson process with rate $\Omega(\theta) + \Omega(\theta^*)$, so that $p(W|\theta, \theta^*) = p(W|\theta, \theta^*)$. The terms $p(X|W, \theta, \theta^*)$ and $p(X|W, \theta^*, \theta)$ are obtained after a forward pass over W using discrete-time transition matrices $B(\theta, \theta^*) = \left(I + \frac{A(\theta)}{\Omega(\theta) + \Omega(\theta^*)}\right)$ and $B(\theta^*, \theta) = \left(I + \frac{A(\theta^*)}{\Omega(\theta) + \Omega(\theta^*)}\right)$.

Calling the parameters after the accept step $(\tilde{\theta}, \tilde{\theta}^*)$, we have that $(\tilde{\theta}, \tilde{\theta}^*, W)$ has the same distribution as (θ, θ^*, W) . Finally, following Lemma 1 in [14], using the matrix $B(\tilde{\theta}, \tilde{\theta}^*)$ to make a backward pass through W , and discarding the self-transitions results in a trajectory $(\tilde{S}(t))$ distributed according to $A(\tilde{\theta})$. Discarding the auxiliary parameter $\tilde{\theta}^*$ results in a pair $(\tilde{\theta}, \tilde{S}(t))$ from the posterior. \square

5.1 Comments

The uniformization scheme of [14] works for any underlying Poisson process whose rate Ω is greater than $\max_i A_i$. The strict inequality ensures that the conditional probability of sampling one or more thinned events U is positive for every trajectory $S(t)$ (recall $U \sim \text{PoissonProc}(\Omega - A_{S(t)})$). Empirical results from [14] suggest setting $\Omega = 2 \max_i A_i$.

Implicit in our new scheme is a uniformizing Poisson process with rate $\Omega(\theta, \theta') = \Omega(\theta) + \Omega(\theta')$. For our scheme to be valid, $\Omega(\theta, \theta')$ must be greater than both $\max_i A_i(\theta)$ and $\max_i A_i(\theta')$. The smallest and simplest such choice is $\Omega(\theta, \theta') = \max A_i(\theta) + \max A_i(\theta')$. For a fixed θ , this reduces to $\Omega = 2 \max A_i$, providing an intuitive motivation for the approach in [14]. Larger alternatives include $\Omega(\theta, \theta') = \kappa(\max A_i(\theta) + \max A_i(\theta'))$ for $\kappa > 1$. These result in more thinned events, and therefore more computation, with the benefit of faster MCMC mixing. We study the effect of κ in our experiments.

It is also possible to have non-additive settings for $\Omega(\theta, \theta')$. A simple option is to set it equal to $\kappa \max(\max_i A_i(\theta), \max A_i(\theta'))$ for some choice of $\kappa > 1$. We investigate this option as well.

6 Experiments

In the following, we evaluate Python implementations of our two proposed algorithms, the naïve MH algorithm (algorithm 4, which we plot in yellow) and its symmetrized improvement (algorithm 5, which we call symmetrized MH and plot in red). We compare different variants of these algorithms, corresponding to different uniformizing Poisson rates (i.e. different choices of κ , see section 5.1). For naïve MH, we set $\Omega(\theta) = \kappa \max_s A_s(\theta)$ with κ equal to 1.5, 2 and 3, represented in our plots with circles, triangles and square symbols. For symmetrized MH, where the uniformizing rate depends on both the current and proposed parameter, we consider two settings $\Omega(\theta, \theta^*) = \kappa(\max A(\theta) + \max A(\theta^*))$ ($\kappa = 1$ and 1.5, plotted with triangles and squares), and $\Omega(\theta, \theta^*) = \kappa \max(\max A(\theta), \max A(\theta^*))$ ($\kappa = 1.5$, plotted with circles). We compare these algorithms against two baselines: Gibbs sampling (algorithm 2, plotted in blue), and particle MCMC [1], plotted in black. Gibbs sampling involves a uniformization step to update the MJP trajectory, and for this we used three settings, $\kappa = 1.5, 2, 3$, plotted with circles, triangles and squares. Unless specified, our results were obtained from 100 independent MCMC runs, each consisting of 10000 iterations. We found particle MCMC to be more computationally intensive, and limited each run to 3000 iterations, the number of particles being 5, 10 and 20 (plotted with circles, triangles and squares).

For each run of each MCMC algorithm, we calculated the effective sample size (ESS) of the posterior samples of the MJP parameters using the R package `rcoda` [12]. This estimates the number of independent samples returned by the MCMC algorithm, and dividing this by the runtime of a simulation gives the ESS per unit time. We used this measure to compare different samplers and different parameter settings.

6.1 A simple synthetic MJP

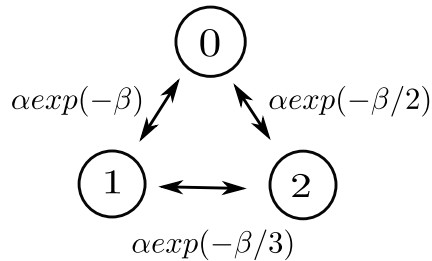


Figure 4: A 3-state MJP with exponentially decaying rates

Consider an MJP with two parameters α and β , transitions between states i and j having rate $\alpha \exp(-\beta/(i+j))$. We consider three settings: 3 states (figure 4), 5 states, and 10 states. We place $\text{Gamma}(\alpha_0, \alpha_1)$, and $\text{Gamma}(\beta_0, \beta_1)$ priors on the parameters α and β , with $(\alpha_0, \alpha_1, \beta_0, \beta_1)$ having values $(3, 2, 5, 2)$ respectively. For each run, we draw random parameters from the prior to construct a transition matrix A , and placing a uniform distribution over states at time 0, simulate an MJP trajectory. We simulate observations uniformly at integer values on the time interval $[0, 20]$. Each observation is Gaussian distributed with mean equal to the state at that time, and variance equal to 1. For the Metropolis-Hastings proposal, we used a lognormal distribution centered at the current parameter value, with a user-specified variance.

Results: Figure 5 plots the ESS per unit time for the parameters α (left) and β (right) for the case of 3 states (top row) and 10 states (bottom row) as we vary the scale-parameter σ^2 of the lognormal proposal distribution. We include results for 5 states in the supplementary material, the conclusions are the same. We see that our symmetrized MH algorithm is significantly more efficient than the baselines over a wide range of choices of σ^2 , (including the natural choice of 1). Among the three setting of our algorithm, the simple additive setting (triangles) does best, though it is only slightly better than the max-of-max setting (circles). A possible reason for this improvement is that the additive setting is more stable than the max-of-max setting, when the proposal variance can be large. The additive setting with a multiplicative factor of 1.5 (squares) does worse than both additive choice with smaller multiplicative factor and the max-of-max choice but still better than the other algorithms. Among the baselines, simple Gibbs sampling does better than naïve Metropolis-Hastings, suggesting that the dependency of the Poisson grid on the MJP parameters does indeed significantly slow down mixing. Particle MCMC has the worst performance for this

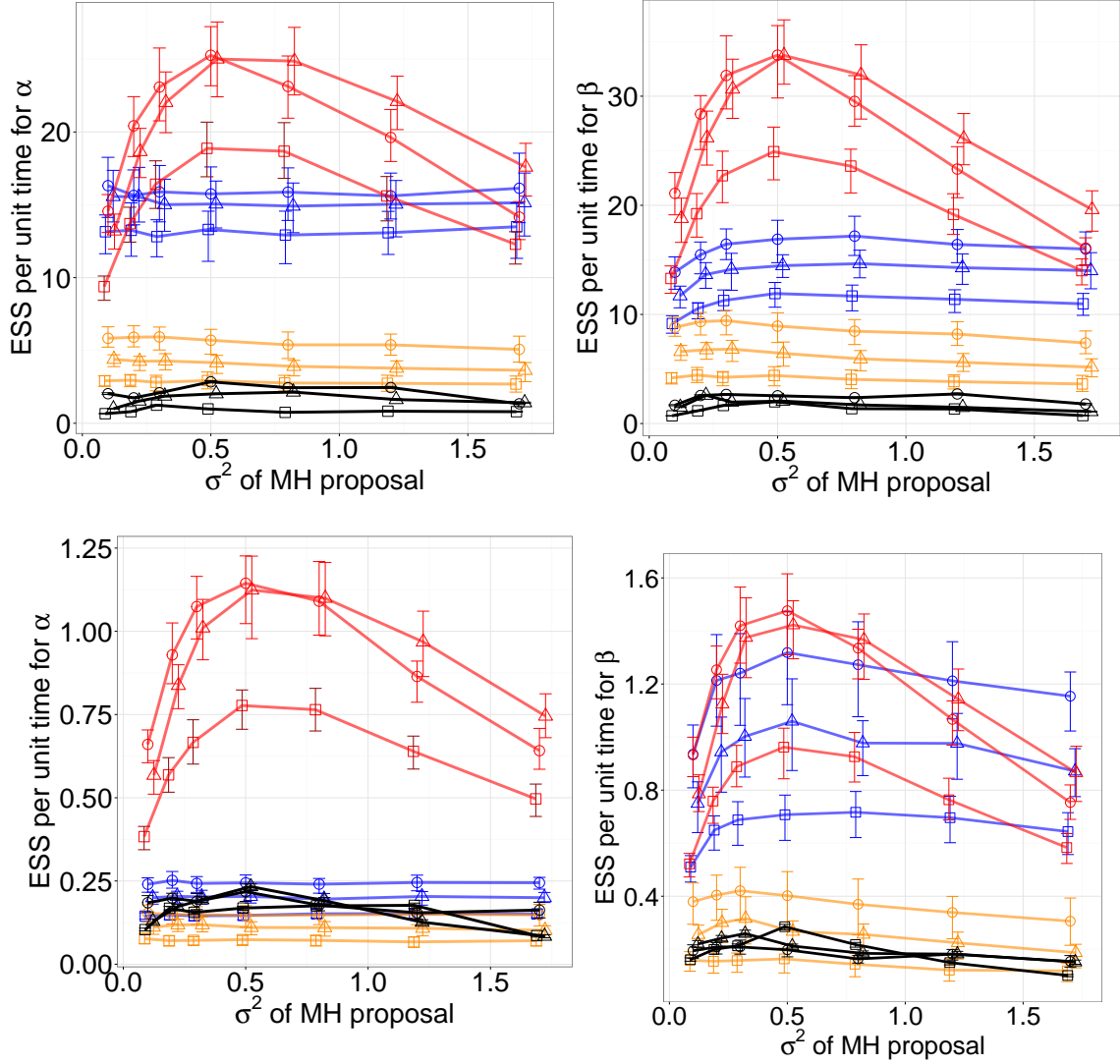


Figure 5: ESS/sec for the synthetic model, the top row being dimension 3, and the bottom, dimension 10. The left column is for α , and the right is for β . Red, yellow, blue and black curves are the symmetrized MH, naïve MH, Gibbs and particle MCMC algorithm. Different symbols correspond to different settings of the algorithms, see section 6

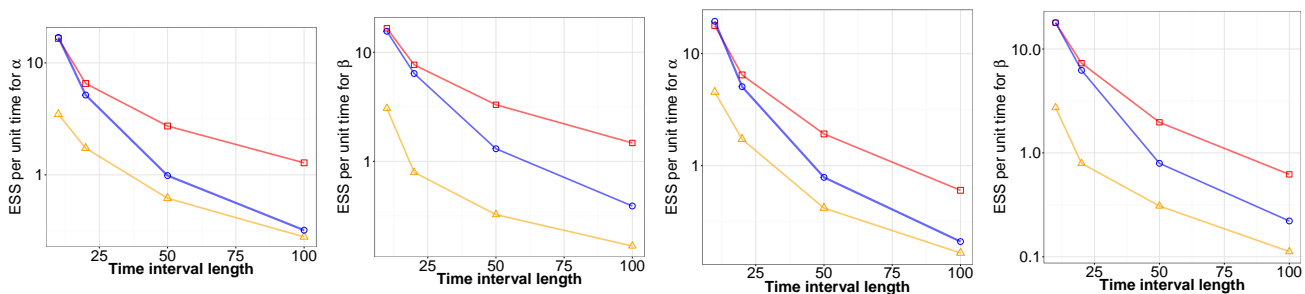


Figure 6: Time Interval vs. ESS/sec. In the left two plots, the number of observations is fixed, in the right two, this grows linearly with the interval length. Red, yellow and blue curves are the symmetrized MH, naïve MH and Gibbs algorithm.

task. The results in figure 5 for the 10-dimensional state space show that for the parameter α , the improvement that our proposed sampler affords is even more dramatic. For the parameter β however, it's performance is comparable to Gibbs, although it's not possible to claim one is uniformly superior to the other.

In figure 6, we plot ESS per unit time as the observation interval \mathcal{T} increases. We consider the three-state MJP, and as before there are 19 observations uniformly located over a time interval $(0, \mathcal{T})$. We consider four settings, with \mathcal{T} equal to 10, 20, 50, 100. For each, we compare our symmetrized MH sampler (with κ set to 1) with the Gibbs sampler (with κ set to 2). While the performance of the Gibbs sampler is comparable with our symmetrized algorithm for the smallest value of \mathcal{T} , its performance is considerably worse for longer time-intervals. This is because of the conditional nature of the updates of the Gibbs sampler, where MJP trajectories are sampled as intermediate objects to facilitate updating the parameters. Longer time intervals will then result in stronger coupling between MJP path and parameters, slowing down mixing. This effect disappears if we integrate out the MJP trajectory. This experiment demonstrates that it is not sufficient just to integrate out the state values of the trajectory, instead, we also have to get around the effect of the trajectory transition times. Our symmetrized MH-algorithm allows us to do this.

In figure 6, we plot results from a similar experiment. Now, instead of keeping the number of measurements fixed as we increase the observation interval, we keep the observation rate fixed at one observation every unit interval of time, so that longer observation intervals have larger number of observations. The results are similar to the previous case: Gibbs sampling performs well for

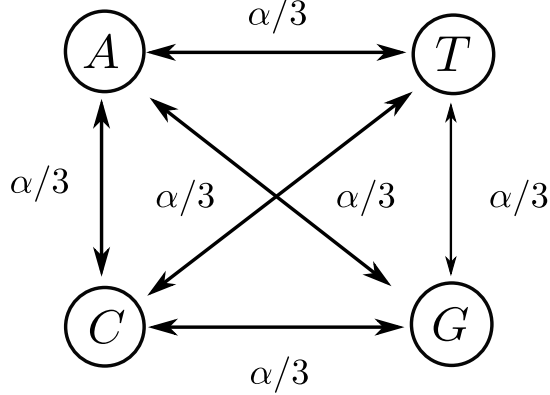


Figure 7: The Jukes and Cantor (JC69) model

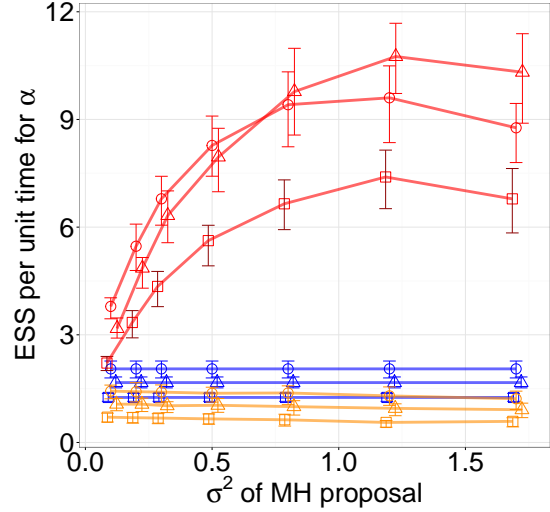


Figure 8: ESS/sec for the JC69 Model. Red, yellow and blue curves are the symmetrized MH, naïve MH and Gibbs algorithm.

small observation intervals, with performance degrading sharply for larger observation intervals. These two experiments illustrate the usefulness of our idea of integrating out the MJP path while carrying out parameter inference.

6.2 The Jukes and Cantor (JC69) model

The Jukes and Cantor (JC69) model is a popular model of DNA nucleotide substitution. We write its state space as $\{0, 1, 2, 3\}$, representing the four nucleotides $\{A, T, C, G\}$. The model has a single parameter α , representing the rate at which the system transitions between any pair of states. Thus, the rate matrix A is given by $A_i = -A_{i,i} = 3\alpha, A_{i,j} = \alpha, i \neq j$. We place a $\text{Gamma}(3, 2)$ prior on the parameter α . Figure 8(right) compares different samplers: we see that the symmetrized MH samplers comprehensively outperforms all others. Part of the reason why the difference is so dramatic here is because the transition matrix is no longer sparse in this example, implying a stronger coupling between MJP path and parameter α . We point out that for Gibbs sampling, the conditional parameter update is conjugate, and there is no proposal distribution involved (hence its performance remains fixed along the x-axis). Particle MCMC performs worse

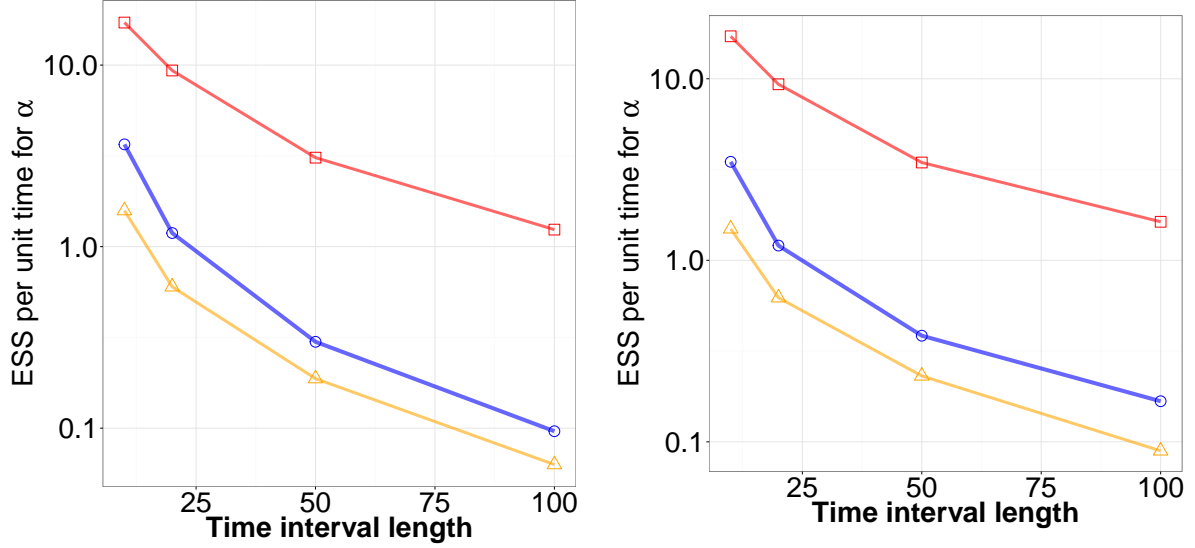


Figure 9: Time Interval vs. ESS/sec. In the left plot, the number of observations is fixed, in the right, this grows linearly with the interval length. Red, yellow and blue curves are the symmetrized MH, naïve MH and Gibbs algorithm.

than all the algorithms, and we do not include it in our plots.

In figure 9, we plot the ESS per unit time for the different samplers as we increase the observation interval. In the left plot, we keep the number of observations fixed, in the right, these increase with the observation interval. Once again we see that our proposed algorithm 1) performs best over all interval lengths, and 2) suffers a performance degradation with interval length that is much milder than the other algorithms.

6.3 An immigration model with finite capacity

Finally, we consider an M/M/N/N queue. This is a stochastic process whose state space is the set $\{0, 1, 2, 3, \dots, N - 1\}$ with elements giving the number of customers/jobs/individuals in a system/population. Arrivals follow a rate- α Poisson process, moving the process from state i to $i + 1$ for $i < N$. The system has a capacity of N , so any arrivals when the current state is N are discarded. Service times or deaths are exponentially distributed, with a rate that is now

state-dependent: the system moves from i to $i - 1$ with rate $i\beta$.

We follow the same setup as the first experiment: for $(\alpha_0, \alpha_1, \beta_0, \beta_1)$ equal to $(3, 2, 5, 2)$, we place $\text{Gamma}(\alpha_0, \alpha_1)$, and $\text{Gamma}(\beta_0, \beta_1)$ priors on α , β . These prior distributions are used to sample transition matrices A , which, along with a uniform distribution over initial states, are used to generate MJP trajectories. We observe these at integer-valued times according to a Gaussian observation process. We consider three settings: 3, 5 and 10 states, with results from 5 steps included in the supplementary material.

Figure 10 plots the ESS per unit time for the parameters α (left) and β (right) as we change the variance of the proposal kernel, for different settings of different algorithms. The top row shows results for a state-space of dimension 3, and the bottom row, results for a dimension 10 (we include the case of dimension 5 in the supplementary material). Again, our symmetrized MH algorithm does best for dimensions 3 and 5, although now Gibbs sampling performs well for dimensionality 10. This is partly because for the problem, the Gibbs conditionals over α and β are conjugate, and have a very simple Gamma distribution (this is also why the Gibbs sampler curves are straight lines: there is no proposal distribution involved here).

6.3.1 A time-inhomogeneous immigration model

Here we extend the previous model to incorporate a known time-inhomogeneity. The arrival and death rates are no longer constant, and are instead given by $A_{i,i+1}(t) = \alpha w(t)$ ($i = 0, 1, \dots, N - 1$) respectively. While it is not difficult to work with sophisticated choices of $w(t)$, here we limit ourselves to a simple piecewise-constant choice of $w(t)$ given by $w(t) = \lfloor \frac{t}{5} \rfloor$. Even such a simple change in the original model can dramatically affect the performance of the Gibbs sampler.

The top row of figure 11 plots the ESS per unit time for the parameters α (left) and β (right) for the immigration model with capacity 3. Now, the symmetrized MH algorithm is significantly more efficient, comfortably outperforming all samplers (including the Gibbs sampler) over a wide range of settings. Figure 11 shows performance for dimension 10, once again the symmetrized MH-algorithm performs best over a range of settings of the proposal variance. We note that increasing the dimensionality of the state space results in a more concentrated posterior, shifting the optimal setting of the proposal variance to smaller values.

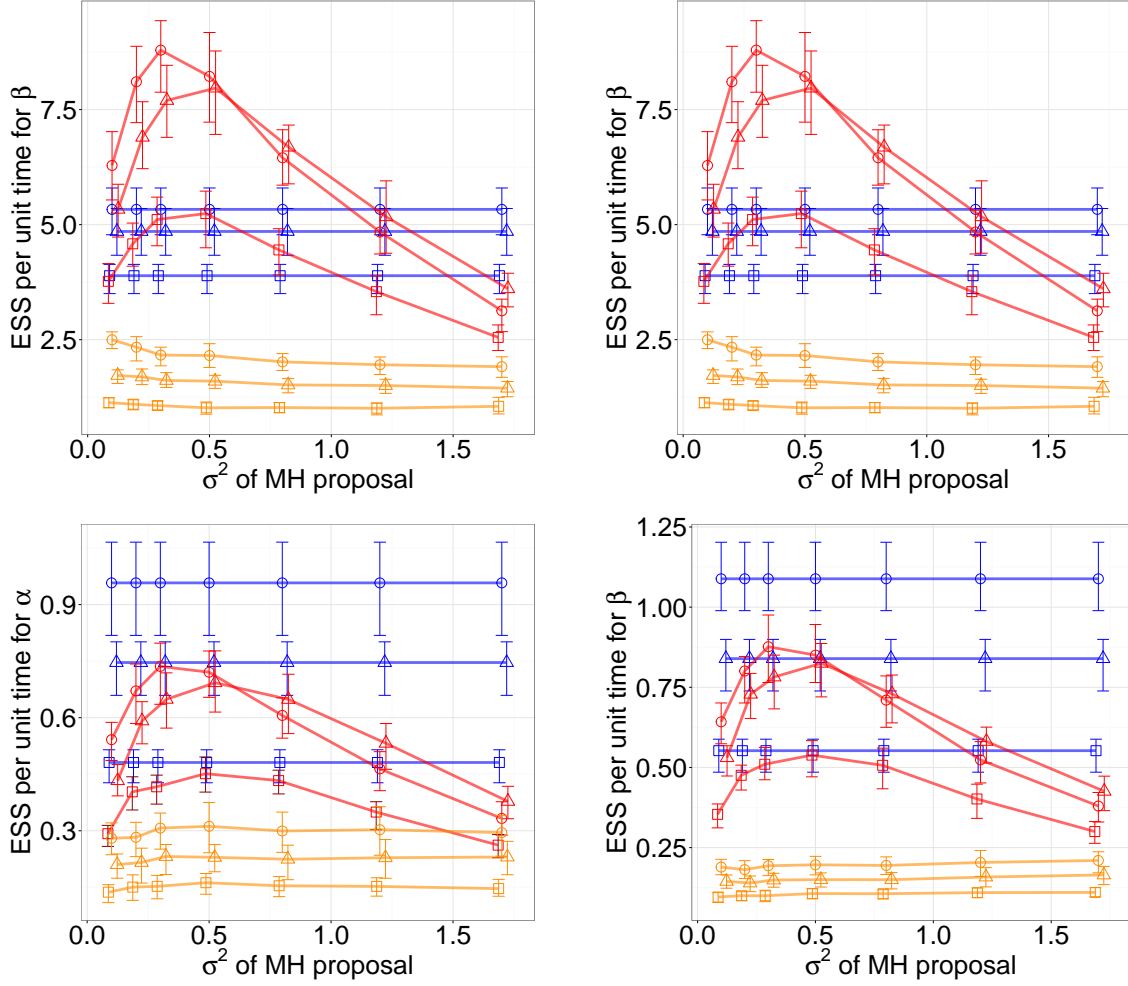


Figure 10: ESS/sec for the immigration model, the top row being dimension 3, and the bottom, dimension 10. The left column is for α , and the right is for β . Red, yellow, and blue curves are the symmetrized MH, naïve MH, Gibbs sampling and particle MCMC.

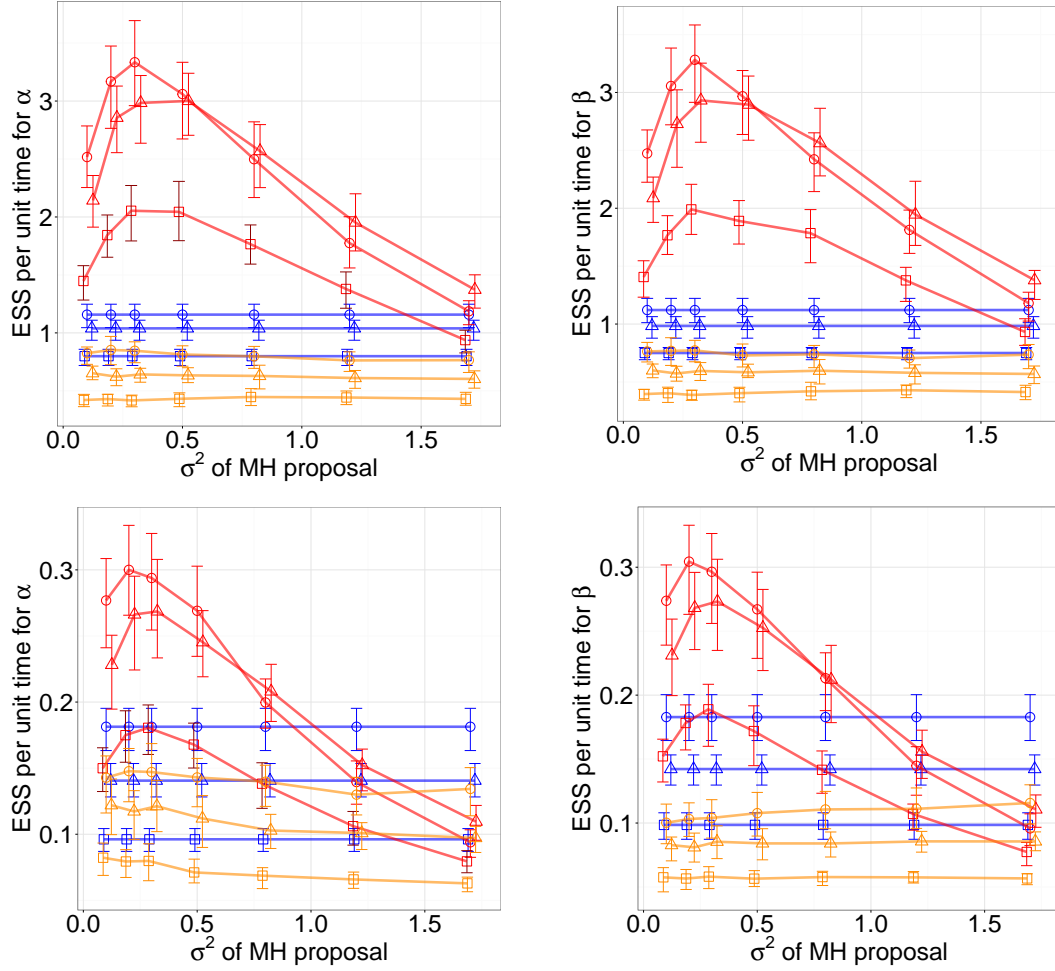


Figure 11: ESS/sec for the time-inhomogeneous immigration model, the top row being dimension 3, and the bottom, dimension 10. The left column is for α , and the right is for β . Red, yellow and blue curves are the symmetrized MH, naïve MH, and Gibbs algorithm.

7 Conclusion

We have proposed a novel Metropolis-Hastings algorithm for parameter inference in Markov jump processes. We use a representation called uniformization to update the MJP parameters with state-values marginalized out, although still conditioning on a random Poisson grid. The distribution of this grid depends on the MJP parameters, significantly slowing down MCMC mixing. We propose a novel symmetrization scheme to get around this dependency. In our experiments, we demonstrate the usefulness of this scheme, which outperforms a number of competing baselines.

There are a number of interesting directions for future research. Our focus was on Metropolis-Hastings algorithms for parameter inference, and though we briefly considered Hamiltonian Monte Carlo, it is interesting to more thoroughly investigate how our ideas extend to this, and other schemes like slice sampling [10]. Another direction is to develop and study similar schemes for more complicated hierarchical models like mixtures of MJPs or coupled MJPs. From a theoretical viewpoint, it is important to complement our empirical studies with theoretical analyses of the mixing properties of our proposed scheme. Finally, we are applying our ideas to other real-world applications from finance and genetics.

References

- [1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, B*, 2010.
- [2] L. Breuer. *From Markov jump processes to spatial queues*. Springer, 2003.
- [3] E. Çinlar. *Introduction to Stochastic Processes*. Prentice Hall, 1975.
- [4] R. Elliott and C. J. Osakwe. Option pricing for pure jump processes with Markov switching compensators. *Finance and Stochastics*, 10:250–275, 2006.
- [5] P. Fearnhead and C. Sherlock. An exact Gibbs sampler for the Markov-modulated Poisson process. *Journal Of the Royal Statistical Society Series B*, 68(5):767–784, 2006.
- [6] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

- [7] N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular biology and evolution*, 11(5):725–736, 1994.
- [8] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skand. Aktuarietiedskr.*, 36:87–91, 1953.
- [9] T. H. Jukes and C. R. Cantor. *Evolution of Protein Molecules*. Academy Press, 1969.
- [10] R. M. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.
- [11] J. Pan, V. Rao, P. K. Agarwal, and A. E. Gelfand. Markov-modulated marked Poisson processes for check-in data. In *Proc. of the 33rd Intern. Conf. on Mach. Learning*, 2016.
- [12] M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, March 2006.
- [13] V. Rao and Y. W. Teh. MCMC for continuous-time discrete-state systems. *In Advances in Neural Information Processing Systems*, (24), 2012.
- [14] V. Rao and Y. W. Teh. Fast MCMC sampling for Markov jump processes and extensions. *Journal of Machine Learning Research*, 2013.
- [15] J. Xu and C. R. Shelton. Intrusion detection using continuous time Bayesian networks. *Journal of Artificial Intelligence Research*, 39:745–774, 2010.

8 Supplementary material

8.1 Synthetic MJP

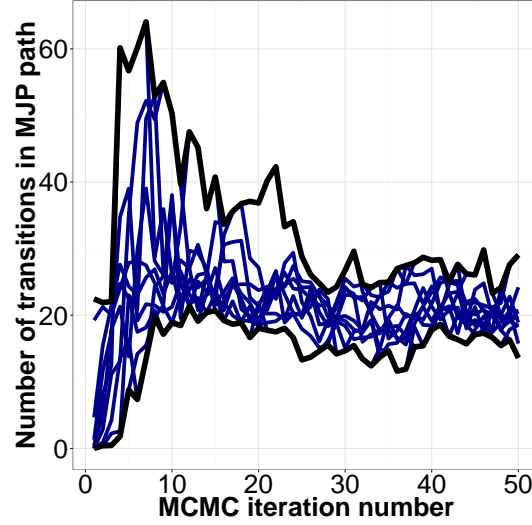


Figure 12: Trace plot of the number of MJP transitions for different initializatoins.

Figure 12 shows the initial burn-in of our improved MH sampler for the immigration-death model for different initializations. The vertical axis shows the number of state transitions in the MJP trajectory of each iteration. This quantity quickly reaches its equilibrium value within a few iterations.

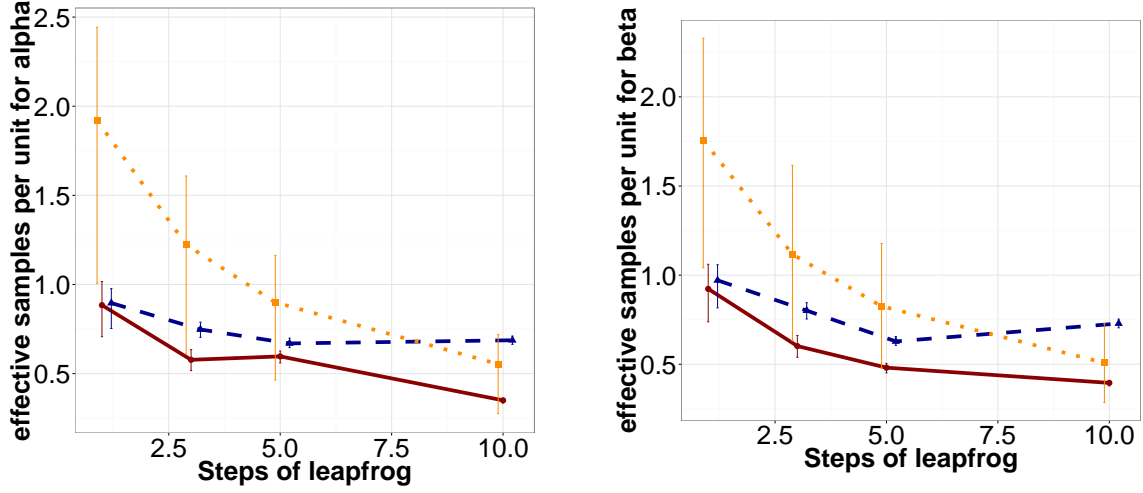


Figure 13: HMC for dim 3

In Figure 13, we plot the ESS per unit time as we change the number of leapfrog jumps in Hamiltonian MCMC for dimension 3 for the immigration-death model. We consider three different step size for leapfrog step($s = 0.02, 0.05, 0.1$). We set the mass matrix M to the identity matrix. We see that in this case, the improved exploration afforded by HMC is not sufficient to overcome the computational burden it incurs: this is partly because every time a gradient is computed (and this every leapfrog step), one needs to run the forward-backward algorithm.

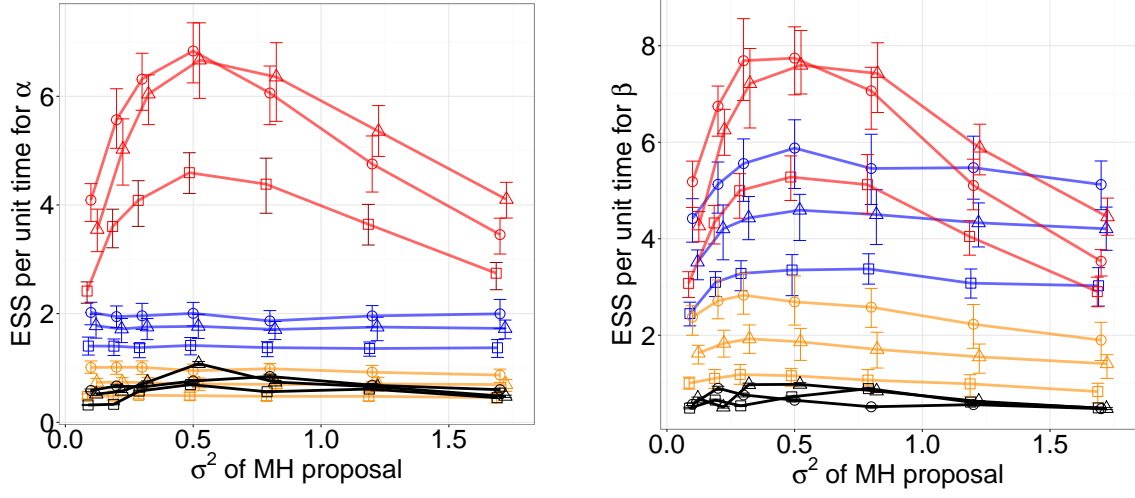


Figure 14: ESS/sec for the synthetic model with dimension 5. The left is for α , and the right is for β .

8.2 Immigration model with capacity

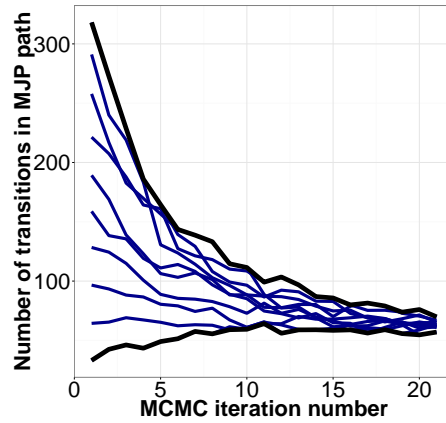


Figure 15: Trace plot of the number of MJP transitions for different initializations for immigration model.

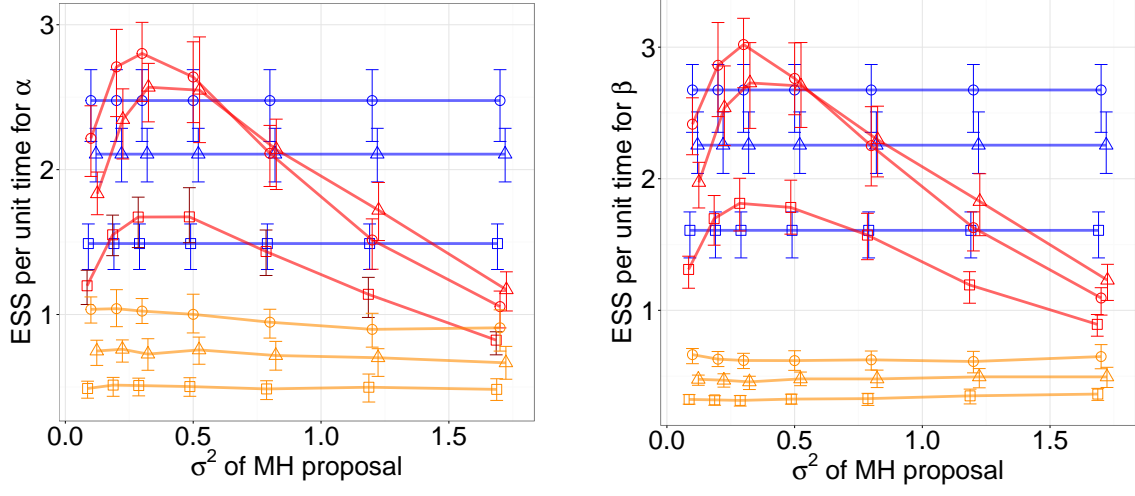


Figure 16: ESS/sec for Immigration model (dim 5). The left is for α , and the right is for β .

8.3 Non-homogeneous immigration model

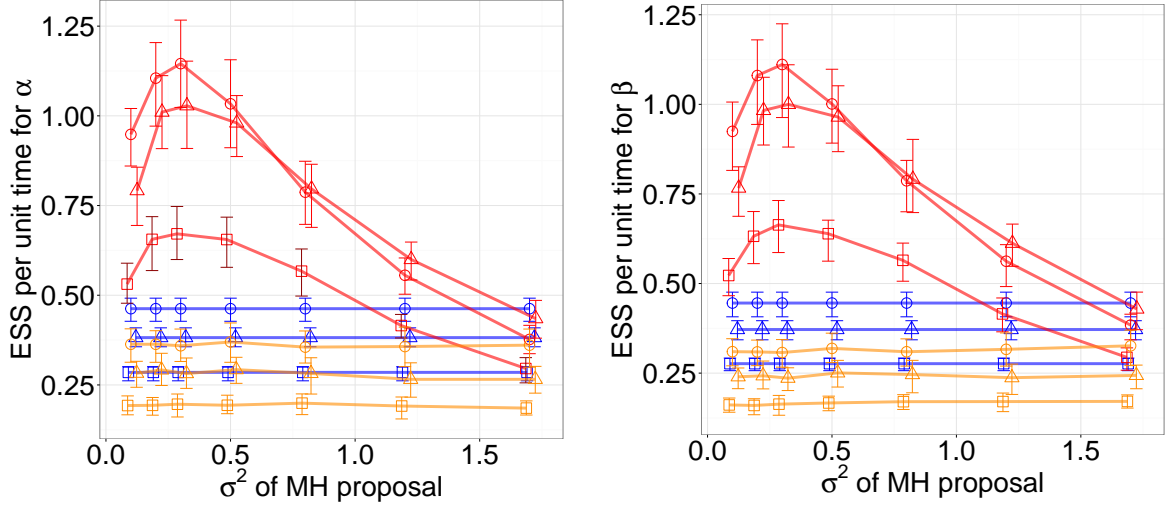


Figure 17: ESS/sec for the nonhomogeneous immigration model (dim 5). The left is for α , and the right is for β .

8.4 Immigration models with capacity

Below we include derivations of the posterior update rules for the immigration model with capacity. Assume our current state consists of a trajectory $S(t) \equiv (S, T)$, with: $S = [S_0, S_1, \dots, S_n]$, $T = [t_0(t_{start}), t_1, \dots, t_n, t_{n+1}(t_{end})]$, and y as observations.

Recall the definition of the immigration model. The state space is $\{0, 1, 2, \dots, N-1\}$, representing the total population. The transition matrix is defined as follows.

$$\begin{aligned} -A_i &=: A_{i,i} = -(\alpha + i\beta), \quad i = 0, 1, \dots, N-1; \\ A_{i,i+1} &= \alpha, \quad i = 0, 1, \dots, N-2; \\ A_{i,i-1} &= \beta, \quad i = 1, \dots, N-1. \end{aligned}$$

And all the other elements are 0. The conditional density(given α, β) of a MJP trajectory

(s_0, S, T) in time interval $[t_{start}, t_{end}]$, with $S = (s_1, s_2, \dots, s_n)$, $T = (t_1, t_2, \dots, t_n)$ is

$$f(s_0, S, T|\alpha, \beta) = \prod_{i=0}^{n-1} A_{s_i, s_{i+1}} \exp\left(\sum_{i=0}^n A_{s_i} (t_{i+1} - t_i)\right).$$

Define

$$U(s_0, S, T) := \sum_{i=0}^{n-1} \mathbb{I}_{\{s_{i+1}-s_i=1\}};$$

$$D(s_0, S, T) := \sum_{i=0}^{n-1} \mathbb{I}_{\{s_{i+1}-s_i=-1\}}.$$

Let us call them U and D for short. Denote the total time when the trajectory state stays at state i as τ_i , i.e. $\tau_i = \sum_{j=0}^n (t_{j+1} - t_j) \mathbb{I}_{\{s_j=i\}}$, then $\sum_{i=0}^n (t_{i+1} - t_i) s_i = \sum_{i=0}^{N-1} \tau_i i$.

$$f(s_0, S, T|\alpha, \beta) = \exp(-\alpha(t_{end} - t_{start} - \tau_N)) \alpha^U \cdot \exp\left(-\left(\sum_{i=0}^k (t_{i+1} - t_i) s_i\right) \beta\right) \prod_{i=1}^{N-1} i^{\sum_{j=0}^{k-1} \mathbb{I}_{s_{j+1}=i-1, s_j=i}} \beta^D$$

We place $\text{Gamma}(\mu, \lambda)$ and $\text{Gamma}(\omega, \theta)$ priors on the parameters α and β , and then the posterior distribution $f(\alpha, \beta|s_0, S, T)$ is as follows:

$$f(\alpha, \beta|s_0, S, T) \propto \exp(-(\lambda + t_{end} - t_{start} - \tau_{N-1})\alpha) \alpha^{\mu+U-1} \cdot \exp\left(-\left(\sum_{i=0}^n (t_{i+1} - t_i) s_i + \theta\right) \beta\right) \beta^{\omega+D-1}.$$

Thus, the posterior distributions of α, β are still independent. In particular,

- $\alpha|s_0, S, T$ is $\text{Gamma}(\mu + U, \lambda + t_{end} - t_{start} - \tau_{N-1})$ distributed.
- $\beta|s_0, S, T$ is $\text{Gamma}(\omega + D, \theta + \sum_{i=0}^n (t_{i+1} - t_i) s_i)$ distributed, which is equivalent to $\text{Gamma}(\omega + D, \theta + \sum_{i=0}^{N-1} \tau_i i)$.

Such immigration models have perfectly conjugate posterior distributions when we assign Gamma priors to α and β .